

# Rajalakshmi Engineering College

Name: sriram sathiyaseelan  
Email: 241001268@rajalakshmi.edu.in  
Roll no: 241001268  
Phone: 9487787666  
Branch: REC  
Department: IT - Section 3  
Batch: 2028  
Degree: B.E - IT

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 10\_Q1

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### **Section 1 : COD**

##### **1. Problem Statement**

A city traffic management system needs to track vehicles entering a toll booth. Each vehicle is uniquely identified by its registration number. The system should allow adding vehicles to a record, ensuring that no duplicate registration numbers exist. The vehicles should be stored in a HashSet, which does not guarantee any specific order.

Your task is to implement a program using a HashSet that allows adding vehicle details and displaying the records.

##### ***Input Format***

The first line of input contains an integer N - the number of vehicles.

The next N lines contain details of each vehicle in the format: "RegNumber

OwnerName VehicleType"

1. RegNumber (String) - A unique registration number (Alphanumeric).
2. OwnerName (String) - The name of the vehicle owner.
3. VehicleType (String, Car, Bike, or Truck) - The type of vehicle.

If a vehicle with the same registration number is already present, ignore the duplicate entry.

### ***Output Format***

The output prints the unique vehicle records in any order (since HashSet does not maintain order).

Output format: "RegNumber OwnerName VehicleType"

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 5

KA01AB1234 John Car

MH02CD5678 Alice Bike

DL03EF9012 Bob Truck

TN04GH3456 Mike Car

KA01AB1234 John Car

Output: TN04GH3456 Mike Car

KA01AB1234 John Car

MH02CD5678 Alice Bike

DL03EF9012 Bob Truck

### ***Answer***

```
// You are using Java  
import java.util.*;
```

```
class Vehicle {
```

```
    String regNumber;
```

```
    String ownerName;
```

```
    String vehicleType;
```

```
    Vehicle(String regNumber, String ownerName, String vehicleType) {
```

```
this.regNumber = regNumber;
this.ownerName = ownerName;
this.vehicleType = vehicleType;
}

// Override equals() to compare vehicles based on registration number
@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null || getClass() != obj.getClass())
        return false;
    Vehicle other = (Vehicle) obj;
    return regNumber.equals(other.regNumber);
}

@Override
public int hashCode() {
    return regNumber.hashCode();
}

@Override
public String toString() {
    return regNumber + " " + ownerName + " " + vehicleType;
}
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = Integer.parseInt(sc.nextLine());

        HashSet<Vehicle> vehicles = new HashSet<>();

        for (int i = 0; i < n; i++) {
            String[] parts = sc.nextLine().split(" ");
            String reg = parts[0];
            String owner = parts[1];
            String type = parts[2];
            vehicles.add(new Vehicle(reg, owner, type)); // duplicates ignored
automatically
        }
    }
}
```

```
        for (Vehicle v : vehicles) {  
            System.out.println(v);  
        }  
  
        sc.close();  
    }  
}
```

**Status :** Correct

**Marks :** 10/10